User's guide

Version 1.02
(Last update: December 05, 2011)

Disclaimer

This guide describes the use and outcomes of LDSO software. The authors accept no responsibility for the accuracy (or inaccuracy) of the results obtained with this software.

Please notify fytourn@gwdg.de upon the discovery of bugs. Comments and suggestions are welcome.

The bibliographical reference for this software is:

Ytournel, F. (2008). Linkage disequilibrium and QTL fine mapping in a selected population. PhD thesis, Station de Génétique Quantitative et Appliquée, INRA.

SUMMARY

INTRODUCTION

The use of Linkage Disequilibrium (LD) greatly increased over the last decades. It has become a classical tool for the detection of QTL by using fine-mapping methods (Meuwissen and Goddard, 2000; Boitard et al. 2006) or genome-wide association studies (Hayes et al. 2009, Pausch et al. 2011). LD is also the key point of the genomic selection (Meuwissen et al. 2001). Finally LD is also used for studying the population histories, for instance to evaluate the evolution of the population sizes over the time (Hayes et al. 2003) or the effects of natural and artificial selection through the identification of selective sweeps (Fan et al. 2010). Simulations are necessary to evaluate the new methods that can be developed for any method using LD.

LD is created over the generations by different evolutionary forces as mutation, genetic drift, or selection. It has been shown that the LD structure in small regions is the result of the history of the population a long time ago while the recent history has an impact on LD at longer distances (Hayes et al. 2003). It is thus necessary to subdivide the simulations into two parts called “historical population” (a long time ago) and “pedigree known population” (recent generations). Random mating is assumed all over the evolution of the population. One or two populations can be simulated simultaneously and then joined at some points of the population history (in the case of admixed populations or experimental crosses). This particularly applies for populations that diverged a long time ago, as it may be the case in pig where there seem to have been several different domestication centres (Larson et al. 2005). LDSO is intended for simulating livestock experimental populations, which are mainly pure-bred or one-way crosses. Following up on the pig example, a classical experimental cross (generally a F2 or a BC population) results from the mating of a European population (Large White or Landrace population) with a Chinese one (Meishan for instance). However, one may

also wish to simulate populations diverging from a common initial one or 4-way crosses. For that purpose, LDSO has the possibility of reading information (haplotypes, phenotypes, genetic map or even pedigrees) and to simulate the evolution from this information on.

Mutation and genetic drift are classically considered in all simulation programs in the historical generations, but not selection. It seems however a reasonable hypothesis to think that selection may have been applied in livestock since domestication, at least on some production, reproduction and behavioural traits. LDSO enables simulating phenotypic selection at any time of the population history. In the historical population as well as in the recent generations, the population size can be increased (population expansion) or reduced (bottleneck) in each generation.

In the section "pedigree known population", LDSO enables simulating natural and experimental livestock populations submitted to different evolutionary histories. First, three methods of evaluation of the breeding values are available for the recent generations in LDSO (phenotypic evaluation, BLUP or in accordance to a given accuracy defined by the user for each generation). The method used to select the individuals can be varied over the generations. LDSO accounts for the different population structures between species (e.g. number of offspring per dam) and may thus be adapted to the species one want to simulate. Furthermore, the experimental populations also differ: the grand-daughter design is classical in cattle or small ruminants, especially for mapping studies, whereas crosses are preferentially used in pigs. Finally, many breeds have herdbooks where the pedigree is recorded over even more than 15 generations in some cases. Simulating populations as close as possible to the real ones is of great interest.

The genetic map is also a key parameter. Depending on the objective of a study, one may need to simulate a whole genome or only a small part. Several kinds of genetic maps can thus

be simulated. While genomic selection or genome-wide scans require simulating complete genomes with a very large number of markers that can be placed randomly, fine-mapping methods are classically used on smaller regions where one may want to place the markers to mimic a concrete situation. Both situations are possible with LDSO.

Generally, in each generation, the individuals are first subject to selection if $s<1$. Once the potential parents have been identified, the actual ones are arbitrarily chosen and mated at random. The principle of gene dropping (MacCluer et al. 1986) makes it possible to apply selection in each generation, contrary to a coalescent process. Each generation is created by transmitting the parental gametes to the offspring following the Mendelian rules. A gamete is formed under the assumption of Haldane's mapping function: a) the number of recombinations is drawn from a Poisson distribution with parameter the length of the chromosome, and b) the crossing-overs to form the gamete that will be transmitted to one descendant are then positioned at random on the chromosome, i.e. assuming no interference. Then mutation can occur. Each locus (marker or QTL) can be allowed to mutate at any time of the simulation (historical and recent generations). Mutation can occur following a stepwise mutational model (Kimura and Ohta, 1978) or a recurrent mutation model to establish a mutation-drift equilibrium in the historical generations. The stepwise mutational model corresponds to microsatellite markers, the DNA Polymerase adding or suppressing a repetition of the repeated sequence, creating an allele different from the parental one. The recurrent mutation model corresponds to the Single Nucleotide Polymorphisms (SNP) where only 2 alleles are coexisting and mutation makes one allele alter into the other. Mutation is follows a binomial distribution at each locus and the haplotypes where mutation happens are drawn at random in the population.

A replicate starts by providing the simulation programme the description of the historical population (file “pop1”) and the genetic map (file “general”). In the historical population file are specified among other the size of the population, the selection rates in the population, the number of generations, etc. The number of markers, QTL, the kind of map etc. should be provided in the genetic map file. A further file (“param”) is required to provide the seeds to the random number generator. Then, for each historical generation, the programme determines the parents of the next generation. If selection applies, the parents are drawn at random among the best individuals (individuals with the best phenotypic value). A different selection rate can be applied to the males and the females. If no selection is applied, all individuals can become parents. Then the haplotypes of the chosen parents are transmitted to the new generation (which completely replaces the parental one) with possible recombinations and mutations. For these generations, LDSO can provide optional information about some indicators on the population such as allelic frequencies, inbreeding coefficients or the LD in the population. For the estimation of the inbreeding coefficients, the founder origin of each locus is recorded and a new number is created each time a mutation occurs. This enables also estimating the length of each founder segment for each chromosome of each individual.

After all historical generations have been simulated, the programme reads the file “popfin” which provides information on the recent part of the simulation (“pedigree-known generations”): structure of the population, selection rates, eventually number of generations etc. The populations are then simulated according to the same rules as before (selection, transmission of the haplotypes) but they now can have a given structure. For instance, one can decide to simulate a grand-daughter population or any real population for which one has the pedigree. During these generations, in addition to all optional files possible in the historical part, LDSO prints the phenotypes, genotypes and pedigree of the simulated population.

INFILES

Population information

The populations are simulated in a forward process following the gene-dropping method (MacCluer *et al,* 1986). One or two populations can be simulated. The parameters to describe the populations and the historical generations have to be provided in a file called "pop1". Both populations are thus independent for all of these parameters but they have the same genetic map (information provided in the file "general"). The randomness of the population creation and evolution relies on the 2nd seed provided in the file "parameter". The information to simulate the recent generations ("pedigree known population") should be contained in a file called "popfin". The random elements of this part rely on the 3rd seed in the file "param".

A particularity of LDSO compared to other publically available simulation programs is that it enables getting in a single run the genotypes with the alleles (allelic state) and, optionally, the founder origin of each locus. For the latter, at the beginning of the simulation, each locus in the founders is given a number corresponding to the number of the founder individual it belongs to. These numbers are independent of the allelic state.

Historical generations

Initial allele frequencies

Three different possibilities are offered. The first one (*all_frq*=0) corresponds to the equifrequency for all alleles of a locus. For the second one (*all_frq*=1), the user should provide the frequencies of all alleles except the last one which is computed. In the last case (*all_frq*=2), the allele frequencies are drawn at random from a uniform distribution.

All these options can be supplanted by the combination of the options *qtl_mute* and *altern*. The table *qtl_mute* contains the number of fixed QTL at the beginning of the simulation in each population. If this number is greater than 0 in any of the simulated populations, then it is possible to decide which kind of situation should be considered:

- In the case of a single simulated population, the fixed QTL are chosen at random (*altern*=0). The allele present at the beginning of the simulation is then a neutral or the deleterious one. This situation can also be chosen when two populations are simulated.
- When two populations are simulated:
    - The fixed QTL can be different in both populations. The fixed allele is chosen at random for each population (*altern*=1). It may be any allele.
    - The fixed QTL are monomorphic in both populations. However, the alleles are different alleles in both populations (*altern*=2): the second population presents the favourable allele while the first population has either an intermediate or the deleterious allele. This would be suited for modelling the introgression of a gene in a population.
    - The QTL can randomly be chosen to be fixed for the same allele in both populations (*altern*=3). The allele present at the beginning is then an intermediate or the deleterious one.

When initially fixed QTL are assumed, then the number of alleles initially provided for each of them is only used as a limit for the number of alleles the QTL may have.

Initial disequilibrium

The populations may be created in different ways, depending on the type of initial disequilibrium wished (at least for the genetic markers. For the QTL, the situation will be influenced by the presence of fixed QTL or not). Some fine-mapping methods rely on the hypothesis of a unique allele or causal mutation at the QTL in the founder population (e.g.

Farnir et al. 2002; Boitard et al. 2006) whereas other methods suppose that all alleles are initially equifrequent (e.g. Meuwissen and Goddard 2000).

No initial disequilibrium (deseq=0)

All alleles are present more than once in the population (according to the initial frequencies wished). The alleles are attributed at random to each individual at each locus. The initial situation is therefore exactly the same for all the simulated populations from an allelic point of view but not for the generated haplotypes as alleles are associated at random.

Complete disequilibrium (deseq=1)

One of the alleles at the QTL position(s) is represented by only one copy in the initial population. The rest of the haplotype is attributed at random for all other loci. The haplotype composed of all the makers (but not the QTL) may not be unique while the whole haplotype (including the QTL) is. Such a situation classically corresponds to a mutation in one of the individuals of a population.

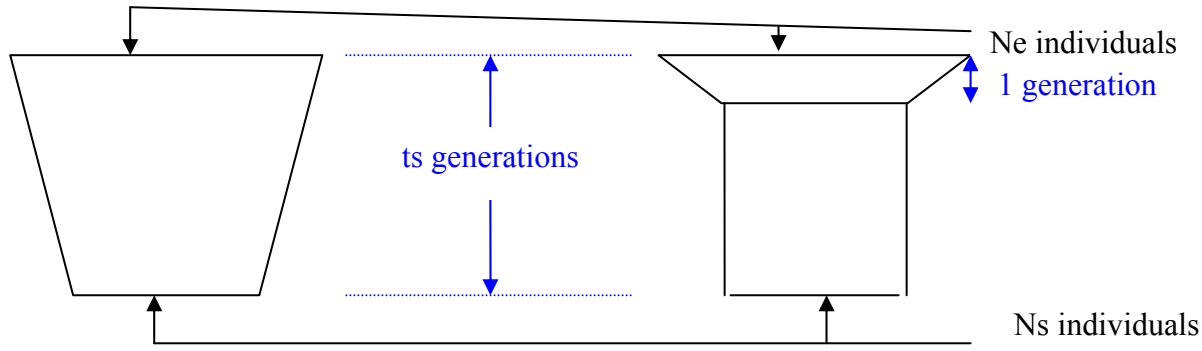
Changes in population size

As many changes as wished can be simulated. Two different ways of changing the population size are possible:

- It occurs on one generation and afterwards the population size remains constant;
- It takes place over more than one generation. The change is then linear over the generations. During the last generation before the reduction or increase of the number of individuals, the population size is Ne; at the end, it is Ns. The change takes place over Ts generations, leading to the equation at the generation t :

  $N(t)=Ne + (Ns-Ne)/t.$

The two possibilities can be summarized by the following scheme:



Progressive change **Sudden change**

Selection

Phenotypes are simulated from the initial founding generation as the sum of the QTL additive effects, eventual dominant and epistatic effects, an infinitesimal additive polygenic effect and an environmental effect. Polygenic Mendelian sampling effects are normally distributed with mean 0 and a variance corrected for each offspring *i* by the inbreeding coefficients of both parents according to the formula $\sigma^2_{Poly_i} = \sigma^2_{Poly} * \left(\frac{2 - (F_s + F_d)}{4} \right)$ where σ^2_{Poly} is the polygenic variance in the founder generation and F_s (resp. F_d) is the inbreeding coefficient of the sire (resp. dam) of the offspring *i*. Residual effects are normally distributed with mean 0 and a variance constant over time. QTL effects can be simulated as additive, dominance and 2-locus epistasic effects. In the case of random effects, the dominance effects are computed as the additive effect times a random number drawn from a uniform distribution between 0 and 1.5. This enables having overdominance cases. Two kinds of epistatic effects described at the molecular level were considered:

- multiplicative effects: the effects of the genotypes at both loci are multiplied (Cordell 2002). The loci A and B have no direct additive effect, but their effect on the phenotype is the product of the overall direct effect of each locus within an individual.

- ”compositional epistasis” (Phillips 2008). The QTL genotype at a locus A is only expressed if the individual has at least one allele with a positive effect at locus B. Biologically, this could correspond to the role of a suppressor, known for instance for qualitative traits such as the blood group.

Different allelic effects can be wished. They are described in the “genomic information” part of this manual. Phenotypes are computed in the historical part as soon as selection is applied. Otherwise, they are computed only in the final populations. The computation of the inbreeding between individuals starts in the first generation even if no selection is applied at that moment.

The simulated populations can be submitted to selection by truncation. The real number seized by the user (between 0 and 1) corresponds to the proportion of the individuals having the best phenotypes in generation *T* that may become parents of the generation *T+1*. When there is simultaneously a progressive change in the population size, the number of potential parents is calculated with the population size at generation *T*.

Simulation from previous data

Since the version 1.02 of LDSO, it is possible to simulate the evolution (historic and/or only in the last generation) of one or two population(s) for which some information is already known. Three possibilities are available:

- only the genetic map and the effects of the QTL are known (*lect_map*=1). The whole population is then simulated conventionally. This enables simulating several populations from the exact initial situation.
- the genotypes, phenotypes and polygenic values of the individuals are already known (*lect_gen*=1). In combination to an already known map, this option enables making a population diverge into several populations evolving independently. This also enables

modelling the mixture of more than two populations if one of the populations is already an admixed population. The genotypes, phenotypes and polygenic values should only be provided for the individuals still alive.

- the pedigree is also known. The simulation then directly goes to the pedigree known part of the simulation.

An example of such a simulation is provided in the example 9.

Table 1: Description of the input parameters in the file "pop1"

Name	Type	Meaning	Comment
nbpop	Integer	Number of populations	1 or 2
nb_bott	Integer	Number of changes in the number of individuals in the 1st population	Should at least be 1
		IF NBPOP = = 2	
nb_bott2	Integer	Number of changes in the number of individuals in the 2nd population	Should at least be 1 On the same line as *nb_bott*
		ENDIF	
n1(0)	Integer	Number of founders of the 1st population	All on the same line
t1(0)	Integer	Number of generations before the 1st change in the population size occurs	
sel(0,1,1:2)	Real	Proportion of the population kept to be parents of the next generation before the 1st bottleneck occurs	
n1(i)	Integer	Number of individuals in the 1st population after the change in the population size	All on the same line for botteleneck *i* Repeat as many times as *nb_bott*
t1(i)	Integer	Number of generations between the population changes *i* and *i*+1	
soud1(i)	Integer	Type of change in the population size: - 1 for a sudden change - 2 for a progressive change	
sel(i,1,1:2)	Real	Proportion of each sex kept to be parents of the next generation during the *i*th change in the population size	
qtl_mute(1:nbpop)	Integer	Number of initially fixed QTL in each population	
		IF AT LEAST ONE QTL IS INITIALLY FIXED	
altern	Integer	Combination of the fixed QTL (see “initial allele frequency” part): =0 : randomly chosen in 1 or 2 populations =1 : chosen at random in both populations with divergent effects =2 : same loci in both populations with divergent effects =3 : same loci in both populations with identical effects	
		ENDIF	

		IF TYP_MK = = 1	
mut(j)	Integer	=1 if the marker is allowed to mutate =0 else	Repeat as many times as *nmq*
		ENDIF	
		IF TYP_MK = = 2	
nball(j,1)	Integer	Number of alleles at each marker locus	Repeat as many times as *nmq*
mut(j)	Integer	=1 if the marker is allowed to mutate =0 else	• Repeat as many times as *nmq* • On the same line as *nball(j,1)*
		ENDIF	
nball(nmq+j,1)	Integer	Number of alleles at each QTL	Repeat as many times as *nqtl*
mut(nmq+j)	Integer	=1 if the QTL is allowed to mutate =0 else	• Repeat as many times as *nqtl* • On the same line as *nball(nmq+j,1)*
h2	Real	Heritability to be simulated	On one line: one value per population
ratio_polQTL	Real	Ratio between the polygenic variance and the additive variance when the phenotypes are generated	On one line: one value per population
		IF NBPOP = = 2	
n2(0)	Integer	Number of founders of the 2nd population	All on the same line
t2(0)	Integer	Number of generations before the 1st change in the population size	
sel(0,2,1:2)	Real	Proportion of each sex kept to be parents of the next generation before the 1st bottleneck occurs	
n2(i)	Integer	Number of individuals in the 2nd population after the change in the population size	• All on the same line for botteleneck *i* • Repeat as many times as *nb_bott*
t2(i)	Integer	Number of generations between the 2 changes in the population size	
soud(i)	Integer	Type of change in the population size: - 1 for a sudden change - 2 for a progressive change	
sel(i,2,1:2)	Real	Proportion of the population kept to be parents of the next generation during the ith change in the population size	
nb_infusion	Integer	Number of introgressions from external individuals in the currently simulated population	

		IF NB_INFUSION > 0	
introg(i)	Integer	Number of individuals that should be used as parents for the introgression	Repeated as many times as *nb_infusion*
time_introg(1:nb_infus,1)	integer	Generations at which individuals should be introgressed	
		ENDIF	
		ENDIF	
		IF LECT_GEN = = 0	
all_frq	Integer	Initial distribution of the allele frequencies: - 0 for a uniform distribution - 1 if all allele frequencies are provided by the user - 2 if the allele frequencies are drawn at random from a uniform distribution	
		IF ALL_FRQ = = 1	
all_fix	Integer	- 0 if all loci are fixed - 1 else	
		If all_fix = = 1	
fq_init(i,1:nball(i,j),j)	Real	Frequency of all alleles except the last one at locus *i* (first for all markers and then for the QTL) in population *j*	Provide all allele frequencies for the 1st population and then, if the case arises, for the second population
		Endif	
		ENDIF	
		ENDIF	

direct_fin	Integer	=0 if an historical population should be simulated =1 else	
		IF DIRECT_FIN = = 1	
old_ped(pop)	Integer	=0 if no pedigree is available =1 else	If two populations are simulated, the values should be on the same line
		IF ANY OLD_PED = = 1	
fic_ped(pop)	Character	Name of the file where the pedigree for the population pop is recorded	If two populations are simulated, the values should be on the same line
indPed(pop)	Integer	Number of individuals in the pedigree	– If two populations are simulated, the values should be on the same line – On the same line as fic_ped(pop)
last_ind	Integer ENDIF	Number of the last individual in the pedigree(s)	
		ENDIF	

Recent population

The "recent" population may be an admixture of the two previous populations or the continuation of the history of the first population (when only one has been simulated previously). LDSO should enable mimicking a wide range of situations, going from natural populations to classical experimental ones. Options *typ_pop*=0, 1 or 2 correspond to simulated populations whereas *typ_pop*=3 and 4 enables following the familiar structure of a known population.

When two populations are simulated, there are two kinds of situations: the population are admixed only once (Fn-populations or random mating) or the second population continue to evolve as a pure breed and there are repetitive admixtures of individuals of the second population into the first one (repetitive introgression or BC-populations). These two situations correspond respectively to the options *typ_pop*=0 or *typ_pop*=2. For both of them, the user must provide on how many generations the pedigree, genotyping (and founder numbers if this has been stated previously) and phenotypes should be recorded. The information of the last historical generation can also be included. The user must strictly define how many males and females are in the population and which proportion of them is kept (with the possibility of having a different selection intensity for the males in the second population when used for breeding with the females of the 1st or the 2nd population if *typ_pop*=2). Concerning the family structure, it can be completely user-provided (through the definition of the number of females per male and offspring per male) or it may occur at random.

If only one population has been simulated, there are two options. Under *typ_pop*=0, the user has the possibility of having a random mating population with the same features as previously. Under the option *typ_pop=1*, the family structure is a grand-daughter design with the possibility of having dams related to each other by specifying the number of maternal

grand-sires. The new population can also be submitted to selection with a new selection rate (defined as previously). The recorded pedigree is composed of the grand-parents and the parents, the recorded phenotypes being the parental ones. The haplotypes and genotypes from the individuals of the 2 last generations or the number of heterozygous sires are also provided. A known pedigree can be used to keep the structure as close as possible to a real one when using the option *typ_pop=3*. It must be stored in a file called "pedig". In this file, the individuals should be listed with continuously increasing numbers, the oldest animals having the lowest numbers.

Finally, one can also make use of a real pedigree to determine the optimal experimental size (*typ_pop=4*). There are two options: simulate news families or add offspring to the current family. The individuals that will be selected as parents could come from the animals in the simulated historical population and (or) from the real pedigree. The total number of animals in the simulated population and in the pedigree must be provided. Concerning the family structure, the user can choose between paternal half-sibs or maternal half-sibs. It is then possible to vary the number of dams and descendant per family (for instance, if you are simulating two families in a population, you may want to have one family with 10 dams and 50 offspring by sire and the other one with 20 dams and 30 offspring by sire). One can also have mixtures of full and half-sib families if a dam has more than 1 offspring and the sire more than 1 dam. The input files from the real pedigree must have the following structures. The genotype file should have one line per individual. The first element is the animal number, followed by the complete first haplotype of the individual (QTL included) and then the whole second haplotype. The pedigree file should also consist of one line per individual with its first element being the individual's number, followed by the corresponding sire number and the dam number. Finally the files containing the phenotypes and breeding values have the same

structure. They should have two columns, the first one showing the individual's numbers and the second one the phenotype or breeding value.

For the final populations corresponding to the types 0 to 2, it is possible to select the parents on a phenotypic base, on their true breeding value (computed as the sum of the polygenic value and the additive QTL effects) corrected or not to have a given accuracy which may be different for both sexes, or on BLUP-estimated breeding values. Moreover, in the types 0 and 2, the mode of selection can be chosen independently for all generations.

Table 2: Description of the input parameters in the file "popfin"

Name	Type	Meaning	Comment
typ_pop	Integer	Type of final population to be simulated: - 0 for an F-population or a population with random mating - 1 for a grand-daughter desing - 2 for a BC-population or repetitive introgressions - 3 for a real pedigree - 4 for a full- or half-sibs design	
		IF TYP_POP = = 0 OR TYP_POP = = 2	
time2	Integer	Number of generations	
		If nbpop = = 2	
cross	Integer	Crossing scheme: - 0 if the population 2 should be introgressed in the 1st population or used or paternal line - 1 if the population 1 should be introgressed in the 2nd population or used or paternal line	
		Endif	
gene_ped	Integer	Number of generations with pedigree	- All on the same line
gene_typ	Integer	Number of generations with genotypes	- These values cannot exceed time2+1
gene_perf	Integer	Number of generations with phenotypes	
		ENDIF	
		IF TYP_POP = = 0	
		If nbpop = = 2	
orig	Integer	Type of population: - 0 for a Fn population - 1 for a random mating population	
		Endif	
sel(1:2,0,1)	Real	Proportion of males and females of the last historical generation selected to produce the first generation of the pedigree-known part	

mode_sel(0,1:2)	Integer	Type of selection for each sex in the 1st historical generation: - 0 for a phenotypic selection - 1 for a selection with a given accuracy of the true breeding values (TBV) (polygene + additive QTL value) - 2 for a selection on BLUP estimated breeding values (EBV)		
n(1:2,i)	Integer	Number of males and females in generation i (i comprised between 1 and time2)		
sel(1:2,i,1)	Real	Proportion of males and females selected in generation *i* to produce the next generation	On the same line as n(1:2,i)	
mode_sel(i,1:2)	Integer	Type of selection for each sex in the historical generation *i*: - 0 for a phenotypic selection - 1 for a selection with a given accuracy of the TBV (polygene + additive QTL value) - 2 for a selection on BLUP EBV		To be repeated on *time2* lines
		IF MAXVAL(mode_sel(g,:)) = = 1	If one of the sexes is selected with a given accuracy in generation *g*	
accur(g,1:2)	Real	Accuracy of the estimation of the TBV in the males (1) and females (2) in generation *g*	To be repeated on as many lines as there are generations selected with a given accuracy	
		ENDIF		
DamSire	Integer	For the attribution of the number of dams per sire - 0 if the parameters are provided by the user - 1 if the attribution should be random		
		IF DAMSIRE = = 0		
constant	Integer	- 0 if the number of dams per sire is kept constant over the generations - 1 else		

evenly	Integer	- 0 if all sires have the same number of dams - 1 else	On the same line as evenly
		IF CONSTANT = = 0	
		If evenly = = 0	
damPerSire(1,1,1)	Integer	Number of dams per sire	
		Endif	
		If evenly = = 1	
damPerSire(1:nt,1,1)	Integer	Number of dams per sire for all *nt* sires that are selected	
		Endif	
		IF CONSTANT = = 1	
		If evenly = = 0	
damPerSire(1,i,1)	Integer	Number of dams per sire in generation *i*	To be repeated on *time2* lines
		Endif	
		If evenly = = 1	
damPerSire(1:nt,i,1)	Integer	Number of dams per sire for all *nt* sires that are selected in generation *i*	To be repeated on *time2* lines
		Endif	
		ENDIF	
		ENDIF	
OffSire	Integer	Attribution of the number of offspring per sire - 0 if the parameters are provided by the user - 1 if this should be random	
		IF OFFSIRE = = 0	
uniformity	Integer	- 0 if all sires have the same number of offspring 1 else	
constant2	Integer	- 0 if the number of offspring per sire is kept constant over the generation - 1 else	On the same line as uniformity

		IF CONSTANT2 = = 0	
		If uniformity = = 0	
offspPerSire(1,1,1)	Integer	Number of offspring per sire	
		Endif	
		If uniformity = = 1	
offspPerSire(1:*nt*,1,1)	Integer	Number of offspring for each of the *nt* sires that are selected	
		Endif	
		ENDIF	
		IF CONSTANT2 = = 1	
		If uniformity = = 0	
offspPerSire(1,i,1)	Integer	Number of offspring per sire in generation *i*	To be repeated on *time2* lines
		Endif	
		If uniformity = = 1	
offspPerSire(1:*nt*,1,1)	Integer	Number of offspring for each of the *nt* sires that are selected in generation *i*	To be repeated on *time2* lines
		Endif	
		ENDIF	
constant3	Integer	- 0 if the number of offspring per dam is constant within a sire - 1 else	
		ENDIF	
		ENDIF	
		IF TYP_POP = = 1	
ngpm	Integer	Number of maternal grand-sires	
nt	Integer	Number of sires	
nf	Integer	Number of dams per sire	
selection(1,1,1:2)	Real	Proportion of males and females of the last historical generation selected to create the 1st generation of the grand-daughter design	

		IF MINVAL(SELECTION(1,1,:)) < 1	
mode_sel(1,1:2)	Integer	Type of selection for each sex in the historical generation *i*: - 0 for a phenotypic selection - 1 for a selection with a given accuracy of the TBV (polygene + additive QTL value) - 2 for a selection on BLUP EBV	
		If maxval(mode_sel(1,:)) = = 1	If one of the sexes is selected with a given accuracy in generation *g*
accur(1,1:2)	Real	Accuracy of the estimation of the TBV in the males (1) and females (2) in generation *g*	
		Endif	
		ENDIF	
		ENDIF	
		IF TYP_POP = = 2	
			In this part, the 1st population is defined as the population where admixture happens, i.e. the 1st historical population if cross=0 or the 2nd if cross=1.
sel(1:4,0,1)	Real	Proportion of males (positions 1 and 3) and females (positions 2 and 4) of the last historical generation in the two populations selected to produce the first generation of the pedigree-known part	
mode_sel(0,1:4)	Integer	Type of selection for each sex in each population in the 1st historical generation: - 0 for a phenotypic selection - 1 for a selection with a given accuracy of the TBV (polygene + additive QTL) - 2 for a selection on BLUP EBV	

n(1:4,i)	Integer	Number of males and females in generation *i* (*i* comprised between 1 and time2)		To be repeated on *time2* lines
sel(1:5,i,1)	Real	Proportion of males and females selected in generation i to produce the next generation. The 5th element is the proportion of males from the 2nd population selected to be crossed with the females of the 1st population.	On the same line as n(1:2,i)	
mode_sel(i,1:2)	Integer	Type of selection for each sex in the historical generation *i*: - 0 for a phenotypic selection - 1 for a selection with a given accuracy of the TBV (polygene + additive QTL value) - 2 for a selection on BLUP EBV	On the same line as n(1:2,i)	
		IF MAXVAL(mode_sel(g,:)) = = 1	If one of the sexes is selected with a given accuracy in generation *g*	
accur(g,1:2)	Real	Accuracy of the estimation of the TBV in the males (1) and females (2) in generation *g*	To be repeated on as many lines as there are generations selected with a given accuracy	
		ENDIF		
nb_introg	Integer	Number of generations where males of the 2nd population should be mated to females of the 1st population		
geneIntrog(i)	Integer	Generation of the i-th admixture		
indIntrog(geneIntrog(i))	Integer	Number of individuals resulting from the i-th admixture	On the same line as geneIntrog(i)	To be repeated on *nb_introg* lines
DamSire	Integer	For the attribution of the number of dams per sire - 0 if the parameters are provided by the user - 1 if the attribution should be random		
		IF DAMSIRE = = 0		
constant	Integer	- 0 if the number of dams per sire is kept constant over the generations - 1 else		
evenly	Integer	- 0 if all sires have the same number of dams - 1 else	On the same line as evenly	

		IF CONSTANT = = 0	
		If evenly = = 0	
damPerSire(1,1,1)	Integer	Number of dams per sire	
		Endif	
		If evenly = = 1	
damPerSire(1:nt,1,1)	Integer	Number of dams per sire for all *nt* sires that are selected	
		Endif	
		IF CONSTANT = = 1	
		If evenly = = 0	
damPerSire(1,i,1)	Integer	Number of dams per sire in generation *i*	To be repeated on *time2* lines
		Endif	
		If evenly = = 1	
damPerSire(1:nt,i,1)	Integer	Number of dams per sire for all *nt* sires that are selected in generation *i*	To be repeated on *time2* lines
		Endif	
		ENDIF	
		ENDIF	
OffSire	Integer	Attribution of the number of offspring per sire - 0 if the parameters are provided by the user - 1 if this should be random	
		IF OFFSIRE = = 0	
uniformity	Integer	- 0 if all sires have the same number of offspring - 1 else	
constant2	Integer	- 0 if the number of offspring per sire is kept constant over the generation - 1 else	On the same line as uniformity
		IF CONSTANT2 = = 0	
		If uniformity = = 0	
offspPerSire(1,1,1)	Integer	Number of offspring per sire	
		Endif	

		If uniformity = = 1	
offspPerSire(1:*nt*,1,1)	Integer	Number of offspring for each of the *nt* sires that are selected	
		Endif	
		ENDIF	
		IF CONSTANT2 = = 1	
		If uniformity = = 0	
offspPerSire(1,i,1)	Integer	Number of offspring per sire in generation *i*	To be repeated on *time2* lines
		Endif	
		If uniformity = = 1	
offspPerSire(1:*nt*,1,1)	Integer	Number of offspring for each of the *nt* sires that are selected in generation *i*	To be repeated on *time2* lines
		Endif	
		ENDIF	
constant3	Integer	- 0 if the number of offspring per dam is constant within a sire - 1 else	
		ENDIF	
		ENDIF	
		IF TYP_POP = = 4	
pedig	Integer	- 1 to create new families - 2 to extend existing families	
		IF PEDIG = = 1	
lect	Integer	- 1 if the males are taken from the real population and the females from the simulated one - 2 if the females are taken from the real population and the males from the simulated one - 3 if all individuals are taken from the real pedigree - 4 if all individuals are taken from the simulated pedigree	

datafile8	String	Name of the file containing the genotypes of the real population	
datafile9	String	Name of the file containing the pedigree of the real population	
datafile10	String	Name of the file containing the phenotypes of the real population	
datafile11	String	Name of the file containing the polygenic breeding valeus of the real population	
ntotped1	Integer	Number of animals in the real pedigree	
nbport	Integer	- 1 if paternal half-sibs should be simulated - 2 if maternal half-sibs should be simulated	On the same line as *ntotped1*
		If nbport = = 1	
nt_hs	Integer	Number of sires	
nfmin_hs	Integer	Minimum number of dams per sire	On the same line as *nt_hs*
nfmax_hs	Integer	Maximum number of dams per sire	On the same line as *nt_hs*
ndmin_hs	Integer	Minimum number of offspring per dam	
ndmax_hs	Integer	Maximum number of offspring per dam	On the same line as *ndmin_hs*
		Endif	
		If nbport = = 2	
nt_hf2	Integer	Number of dams	
nfmin_hf2	Integer	Minimum number of sires per dam	On the same line as *nt_hf2*
nfmax_hf2	Integer	Maximum number of sires per dam	On the same line as *nt_hf2*
ndmin_hf2	Integer	Minimum number of offspring per sire	
ndmax_hf2	Integer	Maximum number of offspring per sire	On the same line as *ndmin_hf2*
		Endif	
		ENDIF	
		IF PEDIG = = 2	
lect	Integer	- 1 if the males are taken from the real population and the females from the simulated one - 2 if the females are taken from the real population and the males from the simulated one - 3 if all individuals are taken from the real pedigree	

nsires	Integer	Number of sires	
datafile8	String	Name of the file containing the genotypes of the real population	
datafile9	String	Name of the file containing the pedigree of the real population	
datafile10	String	Name of the file containing the phenotypes of the real population	
datafile11	String	Name of the file containing the polygenic breeding values of the real population	
ntotped1	Integer	Number of animals in the real pedigree	
nbport	Integer	- 1 if paternal half-sibs should be simulated - 2 if maternal half-sibs should be simulated	On the same line as *ntotped1*
		If nbport = = 1	
nt_hs	Integer	Number of sires	
nfmin_hs	Integer	Minimum number of dams per sire	On the same line as *nt_hs*
nfmax_hs	Integer	Maximum number of dams per sire	On the same line as *nt_hs*
ndmin_hs	Integer	Minimum number of offspring per dam	
ndmax_hs	Integer	Maximum number of offspring per dam	On the same line as *ndmin_hs*
		Endif	
		If nbport = = 2	
nt_hf2	Integer	Number of sires	
nfmin_hf2	Integer	Minimum number of sires per dam	
nfmax_hf2	Integer	Maximum number of sires per dam	
ndmin_hf2	Integer	Minimum number of offspring per sire	
ndmax_hf2	Integer	Maximum number of offspring per sire	
		Endif	
		ENDIF	
		ENDIF	

Genetic information

All information needed to create the general genetic map is provided in a file called “general”. This map, i.e. the location of the loci, is common to both populations. The number of a) loci and QTL, b) QTL with dominance effects, c) pairs of QTL with epistatic effects has to be defined by the user.

LDSO allows the user to simulate several QTLs located on different chromosomal regions (which can be unlinked). Two situations are possible:

- The user provides a completely defined map with the marker positions given in Morgans (typ_map==2);
- The map should be random. To enable keeping the same map for different replicates, the randomness of the map and initial genotypes relies on the 1st seed provided in the file “parameter”. In this case, the user needs to specify if the markers should be equally or unequally spaced.

In any of the cases, the map is then provided in the file “map” (see output files).

The effects of the QTL are also defined by the parameters in this file. LDSO enables simulating populations with additive, dominance and epistatic effects between two loci. Dominance and epistatic effects are restricted to QTL with only 2 alleles. The kind of epistatic effects must be defined in the file “general” by means of the option “*typ_inter*” which takes the value 1 for the multiplicative effects (Cordell 2002) and 2 for the ”compositional epistasis” (Phillips 2008).

To establish a mutation-drift equilibrium, the loci are allowed to mutate according to two models:

- the recurrent mutation model, in which the transition probabilities from one allele to the other are equal. This applies to bi-allelic loci.
- the stepwise mutational model (Kimura and Ohta, 1978) which applies to multi-allelic loci such as microsatellites. The transition probability from the current allele to the allele with one less repetition or one more repetition are assumed to be the same, whereas the transition probability to any other allele is null.

Finally, we supposed that the mutation probability at the QTL loci can be different from those at marker loci because, for instance, they correspond to coding regions. Thus a further mutation rate is required for these loci.

Table 3: Description of the input parameters in the file "general"

Name	Type	Meaning	Comment
tot_length	Real	Total length of the genome	
nchrom	Integer	Number of chromosomes	Same line as *tot_length*
nmq	Integer	Number of markers	
nqtl	Integer	Number of QTL	
model	Integer	Model for the QTL effects: =1 if the effects are drawn from a gamma distribution =2 if the effects are provided by the user	
n_qtl_dom	Integer	Number of QTL with a dominance effect	
ncpl_qtl_interac	Integer	Number of pairs of QTL having an epistatic interaction	Same line as *n_qtl_dom*
store_cop	Integer	=1 if the founder origin of each allele should be recorded =2 else	
lect_map	Integer	=0 if a new genetic map should be created =1 if a previously created genetic map should be read	
lect_gen	Integer	=0 if the population should be completely created =1 if the genotypes, phenotypes and true polygenic values are available	Same line as *lect_map*
further_use	Integer	=0 if the numerator relationship matrix should not be printed at the end of the simulation =1 else	Same line as *lect_map*
		IF LECT_MAP = = 0	
		IF MODEL = = 1	
alpha	Real	Shape parameter of the gamma distribution	
beta	Real	Scale parameter of the gamma distribution	Same line as *alpha*
		ENDIF	

		IF MODEL = = 0	
		If n_qtl_dom > 0	
eff(i)	Real	Additive effect (“genotypic value”, Falconer, 1996) of QTL *i*	Repeated as many times as there are QTL
eff_dom(i)	Real	Dominance effect (“genotypic value”, Falconer, 1996) of QTL *i*	Same line as *eff(i)* for QTL *i*
		If n_qtl_dom = = 0	
eff(i)	Real	Additive effect (“genotypic value”, Falconer, 1996) of QTL *i*	Repeated as many times as there are QTL
		Endif	
		ENDIF	
		IF NCPL_QTL_INTERAC > 0	
typ_interac	Integer	Type of interactions =0 if completely at random =1 if the number of each type of interacting pairs is defined by the user =2 if the type of interaction is defined by the user for each pair of QTL	
		If typ_interac = = 1	
n_interac1	Integer	Number of pairs of QTL with a “multiplicative” interaction	
		If typ_interac = = 2	
typ_inter(i)	Integer	Kind of interaction in the *i*th pair of associated QTL	Repeated as many times as there are pairs of associated QTL
		Endif	
		ENDIF	
typ_map	Integer	Kind of genetic map=1 if the map is constructed at random =2 if all the positions of the markers and QTL are provided by the user	

		IF TYP_MAP = = 1	
lgth_chr	Integer	=1 if the length is the same for all chromosomes =2 if the length of each chromosome is given by the user	
		If lgth_chr = = 2	
lg_chrom	Real	Length of each chromosome in Morgans	Length of all chromosomes on the same line
		Endif	
typ_density	Integer	=1 if there should be the same number of markers on each chromosome =2 else	
		If typ_density = = 2	
nloc(k)	Integer	Number of loci on chromosome *k*	Repeated as many times as *nchrom*
		Endif	
evenly	Integer	=1 if the markers are evenly spaced on the chromosomes =2 else	
		IF TYP_MAP = = 2	
max_loc	Integer	Maximum number of loci (QTL + markers) on one chromosome	
nloc(k)	Integer	Number of loci on chromosome *k*	Repeated as many times as *nchrom*
pos_loc(k,i)	Real	Position of locus *i* on chromosome *k*	Same line as *pos_loc(k,i)* for the same *i* and *k*; Repeated as many times as *nloc(k)*
typ_loc(k,i)	Integer	Kind of locus *i* on chromosome *k* (1 for a QTL, 0 for a marker)	Same line as *pos_loc(k,i)* for the same *i* and *k*; Repeated as many times as *nloc(k)*
		ENDIF	
typ_mk	Integer	=1 if all markers are SNP =2 else	
		ELSE IF LECT_MAP = = 1	
typ_mk	Integer	=1 if all markers are SNP =2 else	
file_map	Character	Name of the file where the map is stored	

file_eff	Character	Name of the file where the additive and dominant effects are stored	
max_loc	Integer	Maximum number of loci (QTL + markers) on one chromosome	
		ENDIF	
tmutSNP	Real	Mutation rate of SNP	
tmutMST	Real	Mutation rate of microsatellites	Same line as *tmutSNP*
tmutQTL	Real	Mutation rate at the QTL loci	Same line as *tmutSNP*
deseq	Integer	Type of initial Linkage Disequilibrium =1 if the causal mutations should be unique =0 else	
typing_err	Integer	=1 if genotyping errors should be assumed	
		IF TYPING_ERR = = 1	
errSNP	Real	Error rate in the genotyping of bi-allelic loci	
errMST	Real	Error rate in the genotyping of multi-allelic loci	Same line as *errSNP*
		ENDIF	
		IF LECT_MAP = = 1	
inter_mute(pop)	Character	For each population, name of the file where the interacting and initially fixed QTL are stored	If two populations are simulated, the two names should be on the same line
		ENDIF	

IF LECT_GEN = = 1				
		IF STORE_COP = = 1		
haplo_file(pop)	Character	Name of the file containing the haplotypes for the population pop		As many times as there are populations
copy_file(pop)	Character	Name of the file containing the founder numbers for the population pop	On the same line as *haplo_file(pop)*	
perf_file(pop)	Character	Name of the file containing the phenotypes and true polygenic values for the population pop	On the same line as *haplo_file(pop)*	
lect_nrm(pop)	Integer	=0 if the numerator relationship matrix between the individuals in the population pop is unknown =1 else	On the same line as *haplo_file(pop)*	
		IF LECT_NRM = = 1		
app_file(pop)	Character	Name of the file containing the values of the numerator relationship matrix for the population pop		
		ENDIF		
		ELSE IF STORE_COP = = 0		
haplo_file(pop)	Character	Name of the file containing the haplotypes for the population pop		As many times as there are populations
perf_file(pop)	Character	Name of the file containing the phenotypes and true polygenic values for the population pop	On the same line as *haplo_file(pop)*	
lect_nrm(pop)	Integer	=0 if the numerator relationship matrix between the individuals in the population pop is unknown =1 else	On the same line as *haplo_file(pop)*	
		IF LECT_NRM = = 1		
app_file(pop)	Character	Name of the file containing the values of the numerator relationship matrix for the population pop		
		ENDIF		

Optional files

If and which optional output files should be created must be provided in the file "files".
These files provide information on different parameters during the evolution of the historical population. The considered parameters are: LD between all loci (*simld*), only between the QTL and all loci (markers and QTL) (*simldq*), or as a summary for bins of a user-defined length (*simld_sum*), the allele frequencies at all loci (*simfqall*), the frequencies of the founder numbers at all loci (*simfqcop*), the average inbreeding at each locus (derived form the founder numbers) (*consang*), the length of the founder segments for all loci (*ibd*) or only the segments containing the QTL (*ibdqtl*), the Polymorphism Information Content (PIC, Botstein et al. 1980) (*haplo-PIC*). For a description of the outputs, see the part "optional files" in the section "Outputs".

Table 4: Description of the input parameters in the file "files"

Name	Type	Meaning	Comment
max_gener_eval	Integer	Maximum number of generations for which statistics should be provided	
outfiles	Integer	- 0 if no optional file should be created - 1 else	On the same line as *max_gener_eval*
		IF OUTFILES /=0	
f_dl	Character	- y if the file containing the LD between all loci should be created - n else	
		If f_dl = = y	
j	Integer	Number of generations studied	
gener_eval(1,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_dlq	Character	- y if the file containing the LD between the QTL and all the loci should be created - n else	
		If f_dlq = = y	
j	Integer	Number of generations studied	
gener_eval(10,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_dl_sum	Character	- y if the file containing the summary of the LD for intervals (bins) should be created - n else	
		If f_dl_sum = = y	
j	Integer	Number of generations studied	
gener_eval(8,1:j)	Integer	Generation numbers	On the same line as *j*
bin	Real	Length of the bin (in M) to compute the statistics	
		Endif	
f_ibd	Character	- y if the file containing the length of all founder segments should be created - n else	
		If f_ibd = = y	
j	Integer	Number of generations studied	
gener_eval(2,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_ibdq	Character	- y if the file containing the length of the founder segments containing the QTL should be created - n else	

		If f_ibdq = = y	
j	Integer	Number of generations studied	
gener_eval(3,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_csg	Character	- y if the file containing the average inbreeding at each locus should be created - n else	
		If f_csg = = y	
j	Integer	Number of generations studied	
gener_eval(4,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_hPIC	Character	- y if the file containing the PIC values of the QTL should be created - n else	
		If f_hPIC = = y	
j	Integer	Number of generations studied	
gener_eval(5,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_fqall	Character	- y if the file containing the allele frequencies at all loci should be created - n else	
		If f_fqall = = y	
j	Integer	Number of generations studied	
gener_eval(6,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
f_fqcop	Character	- y if the file containing the frequencies of the founder numbers at all loci should be created - n else	
		If f_fqcop = = y	
j	Integer	Number of generations studied	
gener_eval(7,1:j)	Integer	Generation numbers	On the same line as *j*
		Endif	
		Endif	

OUTFILES

Standard outfiles

simped

This file provides the pedigree for the last generation which is printed as follows:

individual number sire number dam number generation still alive sex

For the founders (individuals appearing 1st in the pedigree in the case they have neither phenotypes nor molecular information), the sire and dam numbers are coded as 0. The file is structured by population if *typ_pop* equals 0 or 2 (and the population is written at the end of each line) and by sire family if *typ_pop* equals 1.

simhaplo

Each line is composed the following way:

i j M_{1ji} M_{1ji} M_{2ji} M_{2ji} … $M_{(nmq+nqtl)ji}$

with i the individual's number

j the haplotype number for the individual *i*

M_{kij} the allele at locus *k* on the j^{th} haplotype of individual *i*

The genotypes at all loci are written in this file. The alleles are coded as numbers.

simhaploNoQTL

Each line is composed the following way:

i j M_{1ji} M_{1ji} M_{2ji} M_{2ji} … $M_{(nmq)ji}$

with i the individual's number

j the haplotype number for the individual i

M_{kij} the allele at locus k on the j^{th} haplotype of individual i

The genotypes at all marker loci are written in this file. The alleles are coded as numbers.

simcop

This file can only be written if the founder numbers are recorded. Each line is composed the following way:

i j M_{1ji} M_{1ji} M_{2ji} M_{2ji} … $M_{(nmq+nqtl)ji}$

with i the individual's number

j the haplotype number for the individual *i*

M_{kij} the founder number at locus *k* on the j^{th} haplotype of individual *i*

The founder numbers at all loci are written in this file. They are coded as numbers.

simcopNoQTL

This file can only be written if the founder numbers are recorded. Each line is composed the following way:

i j M_{1ji} M_{1ji} M_{2ji} M_{2ji} … $M_{(nmq)ji}$

with i the individual's number

j the haplotype number for the individual i

M_{kij} the founder number at locus k on the j^{th} haplotype of individual i

The founder numbers at all marker loci are written in this file. They are coded as numbers.

simperf

The phenotypes are printed in this file. The lines give the following information:

Individual's number phenotype of the individual polygenic value of the individual

heterozygotes

This file is only printed in the case of a grand-daughter design. It contains the number of sires that are heterozygous at the QTL loci. The provided information is:

QTL number number of sires heterozygous at this QTL frequency of the allele of the QTL affecting the phenotype (calculated only among the sires)

genotyp_err

This file is only produced if genotyping errors are assumed. Each line is composed the following way:

Locus number individual's number chromosome true allele

This option is not available for the extension of the population size (typ_pop=4).

Optional files

Two measures of the LD are computed and recorded in some of the following files: the D' (Lewontin, 1964; Hedrick, 1987) and the χ^2' (Yamazaki, 1978).

NB: The files containing information relying on the copy numbers cannot be computed if the situation typ_pop=4 is used (extension of existing populations with simulated individuals).

Table 5: Description of the output of the optional files

Name of the read variable	Name of the output file	Contains
f_dl	simld	− two loci between which LD is computed − D' value − χ^2' value − population size − generation number − is locus 1 a QTL (1 if yes) − is locus 2 a QTL (1 if yes) − distance between the 2 loci (50 means that the 2 loci are on different chromosomes)
f_dlq	simldq	− QTL and locus between which LD is computed (the QTL can be either in the 1st or 2nd place) − D' value − χ^2' value − population size − generation number − distance between the 2 loci (50 means that the 2 loci are on different chromosomes)
f_dl_sum	simld_sum	− 'Intra' (within one chromosome) or 'Inter' (between chromosomes) − Number of the chromosome − If 'intra', then number of the bin; if 'inter', number of the second chromosome − Measure of the LD (D' or χ^2') − Sum of the LD in the bin − Sum of the squared LD in the bin − Frequency of the LD values in classes (0, $0<LD\leq0.1$, $0.1<LD\leq0.2$, …, $0.9<LD<1$, $LD=1$) − Generation
f_ibd	ibd	− number of the individual − haplotype − number of the segment − length of the segment (in cM)
		− −

		–
f_ibdq	ibd_qtl	– number of the individual – haplotype – QTL number – length of the segment (in cM) – frequency of the QTL allele carried by the individual at the current QTL
f_csg	consang	– locus number – average inbreeding at this locus – variance of the inbreeding for this locus – population size – generation
f_hPIC	haplo-PIC	– population size – generation – locus number – number of haplotypes in the population (haplotypes defined without considering the QTL loci) – proportion of remaining haplotypes relatively to the initial number of haplotypes (defined without considering the QTL loci) – number of haplotypes in the population (haplotypes defined considering the QTL loci) – proportion of remaining haplotypes relatively to the initial number of haplotypes (defined considering the QTL loci) – indicator (1 if the locus is a QTL) – PIC of the locus
f_fqall	simfqall	– locus number – allele – allele frequency – population size – generation number
		–
		–
		–
		–

		–
f_fqcop	simfqcop	– locus number – founder number – number of founder numbers remaining at the current lous – founder number frequency – if the locus is a QTL, number of copies corresponding to the favourable allele – population size – generation number

EXAMPLES

The parameter files corresponding to the following examples are provided in the directory "EXAMPLES" of the zip file.

Example 1: A grand-daughter design

A population of an initial size 100 evolves over 100 generations with no change in the population size (this happens because, even if one bottleneck is stated, the number of individuals remains constant. Note that in such a case, it is preferable to require a progressive change in the population size). The male population is selected with a low intensity over the two first generations (proportion selected=0.8). The trait has an heritability of 0.3 with relies on half on the QTL and on the second half on the polygene.

The genetic map is entirely provided by the user with 200 genetic markers and one QTL in position 61. All genetic markers are biallelic and the QTL initially has five alleles. All alleles initially have the same frequency within a locus.

After these 100 generations, a grand-daughter design is simulated. Fifty sires are mated to 100 dams each to provide thus 100 offspring per sire. The dams are originating from 30 maternal grand-sires. The sires and dams are selected relatively to their breeding values estimated with accuracies of 0.9 and 0.5 respectively. Ten and fifty percents of the sires and dams respectively can be selected as parents.

Example 2: An F2 population

Two populations of initial size 500 individuals are simulated over 100 generations. The first population evolves without any change in its size and with a light selection pressure only in

the last 10 generations. On the other hand, the 2nd population gets progressively to a size of 300 between the generations 3 and 82, and its size decreases even more drastically between the generations 83 and 100 by getting from 300 to 90. In generation 83, the males start to be selected with a low intensity. Population 2 is introgressed in population 1 in generation 95 by producing 20 F1-products.

The final population is a F10-cross with the sires of the F1 being chosen in the historical second population and the dams being chosen in the historical first population. All 10 generations consist of 200 individuals (100 females and 100 males) with 10% of the sires and 80% of the dams being chosen to have offspring in the next generation. The number of sires per dam and how many offspring a sire (and thus a dam) has are randomly chosen. The 10 best percents of the sires are selected in the generations 1 and 2 (the individuals in the last historical generation are unselected) according to their phenotypes and afterwards on their breeding values with accuracies of 0.7 to create the 3rd and 4th generation and 0.8 latter. The females are selected from the 1st generation on with 80% of them being selectable. They are selected until generation 4 included on their phenotype and latter on an estimated breeding value with an accuracy of 0.5.

The genetic map consists of 201 genetic markers and 5 QTL. All loci are placed by the user. The QTL have between 5 and 10 alleles in the first population and between 4 and 12 alleles in population 2. The allele frequencies are determined at random.

The outputs consist of:

- the pedigree of the F-population over the last 10 generations,
- the molecular information (alleles and founder origin) for the last 3 generations
- the phenotypes of the last 9 generations
- the length of the founder segments containing the QTL in the last historical generation and the last generation of the recent history

- the LD between the QTL and any other locus at the creation of the population, in the last historical generation and in the last generation of the recent history
- the allele frequencies at all loci in the 50th and in the last generation of the historical part and in last generation of the recent history.

Example 3: A BC population

The evolution of the historical populations is the same as in Example 2. The F2 population is also simulated as previously. After this F2 generation, 4 generations of back-crossing are added with the male parents being chosen in the second population and the female ones in the admixed population.

The genetic map differs from that of the F2 population as 2 fixed QTL are simulated in each population with the option "altern=2", implying that the 2 QTL fixed are the same in both populations and that the alleles have divergent effects (the most favourable allele being fixed in the second population). Genotyping errors and missing genotypes are allowed. The real alleles are provided in the file "genotype_err".

Example 4: A random mating population

The initial population is composed of 200 individuals. The population size remains constant over 500 generations. The final population is composed over 8 generations by 500 males and 1000 females. Only 50 males and 500 females are selected to reproduce. The selected males have between 127 and 1 descendant. The number of offspring per selected dam is chosen at random. All individuals were selected on the basis of their phenotypic value.

The genetic map is composed of 1000 markers and 100 QTL. All loci are initially fixed. The maximum number of alleles per locus that can be achieved is 2. Neither dominance nor epistatic effects are assumed.

Example 5: According to an existing pedigree

The historical part consists of 1000 generations with 2000 individuals. The simulated trait has an initial heritability of 0.5 which is for 30% due to a polygenic component.

The genetic map is composed by 100 markers located on 3 chromosomes having different length (0.15, 0.1 and 0.05 M). There are 10 QTL segregating, 1 of them having a dominance effect and 1 pair of QTL showing epistatic interactions. All QTL are biallelic and all loci are initially fixed. The mutation rate for the biallelic loci was set to 10^{-4}. LD was recorded in generations 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000. The results are presented in the section "LD analysis in a data set".

The final population follows the pedigree provided in the file "pedig". It is composed of 1216 individuals, 610 of them being founders and two animals having an unknown parent.

Example 6: Extending an existing pedigree with new families

The historical population is composed of two populations evolving in total over 100 generations. The first one is subjected to 3 changes in its size, reducing it from 1000 to 68 individuals. A bottleneck also happens in the second population reducing its size from 500 to 100. The first population receives recursive introgressions from individuals of the 2^{nd} population between the generations 15 and 24.

A single QTL is assumed having only an additive effect and 5 alleles. The genetic map is composed of 500 markers which positions are given by the user. The allele frequencies are initially uniformly distributed.

In the recent part, new populations are added to an existing pedigree. The initial pedigree is stored in "fic_genea". It contains 8000 individuals whose genotypes are provided in "fic_geno". Their phenotypes and breeding values are respectively in "fic_perf" and

"fic_gind". The extension is done by creating new paternal half-sibs from 4 sires of the last historical generation having each 3 dams form the real pedigree and 1 offspring per dam.

Example 7: Extending an existing pedigree with new individuals in the families

The same historical population and genetic map were taken to extend the families of a population with a real pedigree. Ten sires are taken for that purpose from the real pedigree and the females are issued from the historical population. As previously, the extension is done by creating new paternal half-sibs from 4 sires of the last historical generation having each 3 dams form the real pedigree and 1 offspring per dam.

Example 8: Using previous data

The files containing the haplotypes, copy numbers and phenotypes consisted of the data from 200 individuals. The pedigree was composed of 1'100 individuals from 5 generations. The genetic map had 1'037 markers and 15 QTL on one chromosome, all QTL having dominance effects.

The population derived from this known population consisted in 5 generations of random admixture.

LD analysis in a data set

The files required to simulate this dataset are provided where the same as those for the example 5. Three chromosomes were simulated with different lengths (0.15, 0.1 and 0.05 M). In total, 110 loci were simulated with the same number of markers on each chromosome. Initially, all markers were fixed.

The LD between all pairs of loci was recorded at generations 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000. Figure 2 represents the overall distribution of the LD values (computed with the r^2) as a function of the distance for the generations 100, 200, 500 and 1000.

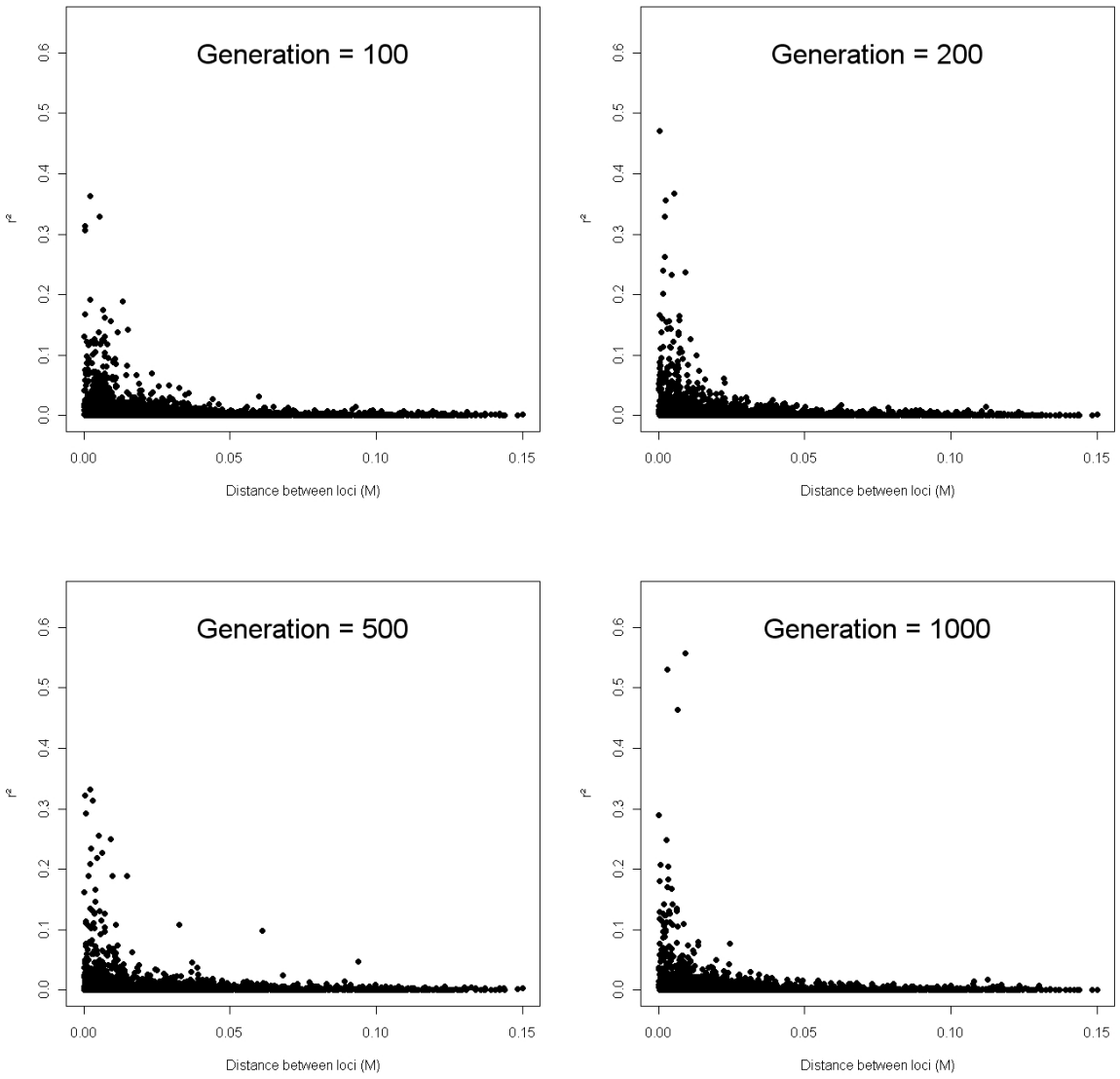


Figure 2: LD (r^2) between all pairs of loci in generations 100, 200, 500 and 1000

Study of the mutation-drift equilibrium

The files required to simulate the scenario with 200 individuals are in the directory "EX8". Two effective population sizes were considered: 200 or 1000 individuals. The populations

evolved over 5'000 generations in total and the allele frequencies were recorded in generations 1'000, 2'000 and 5'000. The genetic map comprised 10'000 markers and 100 QTL. All loci were fixed at the beginning of the simulation. The mutation rate (μ) of the loci (all bi-allelic and thus following a recurrent mutation model) was set to 2.5 10^{-4}. According to Wright (1931) the distribution of the allele frequencies at the mutation-drift equilibrium depends on the value of 4Ne μ. In the case of the smallest population size, this parameter equals 0.2 and the expected distribution should be U-shaped. For the second population size, 4Ne μ=1 and the expected frequency should follow a uniform distribution. The results for 50 replicates are presented in Figure 3.

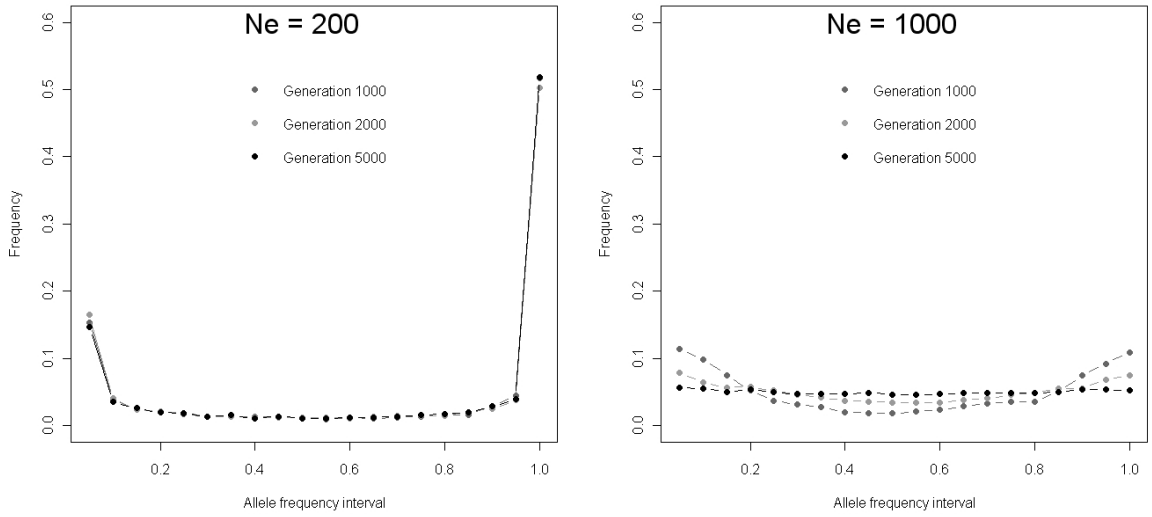


Figure 3: Frequencies of the different allele frequency classes after 1000, 2000 and 5000 generations for populations of effective size 200 and 1000.

REFERENCES:

Boitard S., Abdallah J., de Rochambeau H., Cierco-Ayrolles C., Mangin B., 2006. Linkage disequilibrium interval mapping of quantitative trait loci, BMC, 7: 54.

Botstein D., White R.L., Skolnick M., Davis R.W., 1980. Construction of a genetic map in man using restriction fragment length polymorphisms, Am. J. Hum. Genet., 32:314-331.

Cordell, H.J. (2002) Epistatis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Human Molecular Genetics, 11, 2463-2468.

Fan B., Du Z.-Q., Gorbach D.M., Rotschild M.F., 2010. Development and application of high-density SNP arrays in genomic studies of domestic animals, Asian-Australian Journal of Animal Science, 23(7): 833-847.

Farnir F., Grisart B., Coppieters W., Riquet J., Berzi P., Cambisano N., Karim L., Mni M., Moisio S., Simon P., Wagenaar D., Vilkki J., Georges M., 2002. Simultaneous mining of linkage and linkage disequilibrium to fine map quantitative trait loci in outbred half-sib pedigrees: revisiting the location of a quantitative trait locus with major effect on milk productionon bovine chromosome 14. *Genetics,* 161: 275-287.

Larson G., Dobney K., Albarella U., Fang M., Matisoo-Smith E., Robins J., Lowden S., Finlayson H., Brand T., Willerslev E., Rowley-Conwy P., Andersson L., Cooper A. (2005) Worldwide Phylogeography of Wild Boar Reveals Multiple Centers of Pig Domestication. Science, 307, 1618-1621.

L'Ecuyer, P. (1996) Maximally equidistributed combined Tausworthe generators, Math. of Comput., 65, 203-213, available at http://jblevins.org/mirror/amiller/taus88.f90.

Hayes B.J., Bowman P.J., Chamberlain A.J., van Tassell K.S.C.P., Sonstegard T.S., Goddard M.E., 2009. A Validated Genome Wide Association Study to Breed Cattle Adapted to an Environment Altered by Climate Change, PLoS ONE, 4(8): e6676.

Hayes B.J., Visscher P.M., McPartlan H.C., Goddard M.E., 2003. Novel Multilocus Measure of Linkage Disequilibrium to Estimate Past Effective Population Size, Genome Research, 13: 635-643.

Hayes B. and Goddard M.E., 2001. The distribution of the effects of genes affecting quantitative traits in livestock. Genet. Sel. Evol., 33(3): 209-229.

Hedrick P., 1987. Gametic disequilibrium measures: proceed with caution, Genetics, 117: 331-341.

Kimura M. and Ohta T., 1978. Stepwise mutation model and distribution of allelic frequencies in a finite population. Proc. Natl. Acad. Sci. USA, 75(6): 2868-2872.

Lewontin R.C., 1964. On measures of gametic disequilibrium, Genetics, 49: 49-67.

MacCluer J.W., VandeBerg J.L., Read B., Ryder O.A., 1986. Pedigree analysis by computer simulation, Zoo. Biol., 5: 147-160.

Meuwissen T.H.E., Hayes B.J., Goddard M.E., 2001. Prediction of total genetic value using genome-wide dense marker maps, Genetics, 157: 1819-1829.

Meuwissen T.H.E., Goddard M.E., 2000. Fine mapping of quantitative trait loci using linkage disequilibria with closely linked marker loci, Genetics, 155: 421-430.

Phillips P.C. (2008) Epistasis – the essential role of gene interactions in the structure and evolution of genetic systems. Nature Reviews, 9, 855-867.

Pausch H., Flisikowski K., Jung S., Emmerling R., Edel C., Götz K.-U., Fries R., 2011. Genome-Wide Association Study Identifies Two Major Loci Affecting Calving Ease and Growth-Related Traits in Cattle, Genetics, 187: 289-297.

Yamazaki T., 1977. The effects of overdominance on linkage in a multilocus system, Genetics, 86: 227-236.

License for the random generator

The random number generator incorporated in LDSO is based on a code from L'Ecuyer downloaded from

http://jblevins.org/mirror/amiller/taus88.f90.

L'Ecuyer's 1996 random number generator.

Fortran version by Alan.Miller @ vic.cmis.csiro.au

N.B. This version is compatible with Lahey's ELF90

http://www.ozemail.com.au/~milleraj

Latest revision - 30 March 1999